

02/11/00
JC535 U.S. PTO

PATENT APPLICATION TRANSMITTAL LETTER
(Large Entity)

Docket No.
8700137-0006

TO THE ASSISTANT COMMISSIONER FOR PATENTS

Transmitted herewith for filing under 35 U.S.C. 111 and 37 C.F.R. 1.53 is the patent application of:

DIMARCO, Chrysanne and FOSTER, Mary Ellen

For: **METHOD AND APPARATUS FOR AUTHORIZING OF CUSTOMIZABLE MULTIMEDIA DOCUMENTS**

(This is a continuation of PCT/CA98/00771)

Enclosed are:

- Certificate of Mailing with Express Mail Mailing Label No.
 8 sheets of drawings.
 A certified copy of a application.
 Declaration Signed. Unsigned.
 Power of Attorney
 Information Disclosure Statement
 Preliminary Amendment
 Other:

jc584 U.S. PTO
09/502233
02/11/00

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	12	- 20 =	0	x \$18.00	\$0.00
Indep. Claims	3	- 3 =	0	x \$78.00	\$0.00
Multiple Dependent Claims (check if applicable)	<input type="checkbox"/>				\$0.00
				BASIC FEE	\$690.00
				TOTAL FILING FEE	\$690.00

- A check in the amount of to cover the filing fee is enclosed.
 The Commissioner is hereby authorized to charge and credit Deposit Account No. as described below. A duplicate copy of this sheet is enclosed.
 Charge the amount of as filing fee.
 Credit any overpayment.
 Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
 Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Dated: February 11, 2000

[Handwritten Signature]
Signature
Kevin Pillay Reg. No. 41,559
Orange Chari Pillay
P.O. Box 190, Suite 3600
Toronto Dominion Bank Tower
Toronto-Dominion Centre
Toronto, Ontario M5K 1H6 Canada
Telephone: (416) 868-3457

cc:

A Method and Apparatus for Authoring of Customizable Multimedia Documents

This is a continuation of PCT/CA98/00771 filed August 11, 1998.

This invention relates to a method and apparatus for the authoring of
5 customizable multimedia documents and the adaptive generation of versions thereof
for particular uses.

Background of the Invention

Natural Language Generation (NLG) is a young but growing research field,
10 whose goal is to build computer systems that automatically produce fluent and
effective texts in various human languages. Generally, NLG systems have used
knowledge databases containing general world knowledge and specific domain
knowledge, together with various linguistic resources (e.g., lexicons, grammars,
discourse relations), to produce texts with limited variation in word choice, sentence
15 and discourse structure, and virtually no variation in rhetorical style or pragmatic
purpose.

While various computational systems have been devised as solutions to the
problem of producing documents with limited expressiveness in form and effect, none
has presented a general solution to the problem of representing the kinds of
20 knowledge that are needed to produce documents tailored to a specific use or
audience in a manner that is systematic and extensible, and that further provides for
the authoring of such documents by a non-computer-programmer professional writer.
In addition, no system has yet presented a general solution for automatically
integrating various aspects of document design (e.g., text, graphics, and presentation
25 layout) into a single consistent representation format for use within a document
intended for customization.

It is well-known from studies in communication that presentation of
information in a manner that is tailored to the characteristics of a particular audience
can be significant both in maintaining the interest of the members of the audience and
30 in effectively conveying the meaning of the information. For example, in the health
care industry, it has been shown that information that is tailored to the characteristics
of an individual patient can have a far greater effect in producing compliance with
suggested medical regimens as compared to generic information. (Strecher et al 1994
have done pioneering work in this area).

But as Strecher et al's behavioural studies also showed, a sizeable number of different medical and personality factors had to be taken into account in producing customized health information that would have the desired effect on the intended patient. DiMarco et al (1995) noted that this kind of customization involves much more than producing each brochure or leaflet in half a dozen different versions for different audiences. Rather, the number of different combinations could easily be in the tens of thousands. While not all distinct combinations might need distinct customizations, it is nonetheless impossible in general to produce and distribute, in advance of need, the large number of different editions of each publication that is required for individual tailoring of information.

Thus there is a need for a computer system for the automated production of customized material that would tailor a general-purpose "master document" for a particular purpose or individual on demand. It must also be remembered that in the present context the term "document" is broadly used to define any textual or non-textual data, including multimedia and hypertext, having inter-relationships between the data, and that may be displayable or presented to a human audience in one of many presentations and formats.

As a further example, a master document may refer to the complete superset of instructions to direct the actions of a robot on an assembly line. In this instance, there exists a need to tailor or adaptively generate subsets of combinations of instructions for specific robot applications. Whether the master document is to be customized for a particular purpose, as in the robot example, or tailored for a specific audience, as in the case of health information, this process of adaptive document generation should be easily implementable on a computer system at minimum possible cost and maximum possible ease of use to both the author of the master document and the user of the generation system.

In the field of natural language processing, or, computational linguistics, various computer systems have been implemented which attempt to produce customized documents. In the simplest cases, simple mail-merge techniques are used which enable "personalized" documents to be generated by using hand-coded decision rules indicating what information is to be included for various tailoring situations. However, these techniques result in very inflexible, and often, awkwardly structured, and poorly cohesive texts. Other systems utilize schema-based techniques to select and organize the content data according to simple document-template structures. But

these templates are either too general-purpose to provide anything more than very coarse-grained adaptation in the resulting customized texts or too specific to the application in question to be appropriate for general use in adaptive generation systems.

5 A number of projects have used more sophisticated techniques from NLG research to build adaptive generation systems for both written texts and hypertext documents. The IDAS project (Reiter, Mellish, and Levine 1995) recognized the need to tailor both textual and non-textual information, including visual formatting, hypertext input, and graphics output. IDAS also tried to address the need for explicit
10 authoring tools in the adaptive document generation process, but here the focus was on authoring at the knowledge-base level (i.e., at the level of a computer system's internal representation), while there still exists a need to provide an authoring tool that may be used by a non-computer-programmer professional writer who could compose the master document at the level of ordinary English, with additional markup as
15 required (e.g., HTML markup to support an HTML presentation format for a resulting customized version of the document). IDAS relies mainly on canned texts and aims to provide the user with a means of navigating through the whole "hyperspace" of possible (canned) texts. There is however a need to provide for a much finer-grained degree of tailoring than the IDAS implementation.

20 While IDAS relies mainly on canned texts, other adaptive generation systems do use more-dynamic text generation: the Migraine system (Carenini, Mittal, and Moore 1994) uses an approach to text planning that adaptively selects and structures the information to be given to a particular reader. However, Migraine relies on a large number of context-sensitive and user-sensitive "text plans" (i.e., text schemas)
25 so that its methods of tailoring must of necessity be very specific to its particular domain. The PEBA-II system (Milosavljevic and Dale 1996) uses more-general text plans, as well as text templates, that it can choose from to adapt information to the individual reader, but the tailoring done is very specific, focussing on the user's familiarity with a topic. The PIGLET system (Cawsey, Binsted, and Jones 1995) also
30 uses a combination of text plans and text templates, but its tailoring is also quite specific in nature, mainly concerned with emphasizing material that is relevant to a particular patient. The ILEX-0 system (Knott, Mellish, Oberlander, and O'Donnell 1996) is similar to the PIGLET model in its anticipation of all the possible texts that might be generated, but also includes annotations (e.g., a condition on a piece of

canned text) to allow some local customization. However, very free and flexible use of annotations could lead to problems of repetitive text and inappropriate use of referring expressions in the resulting document, requiring textual repair.

None of the previous systems provide for a text-repair facility of the kind
5 described by Hovy and Wanner (1996) and Wanner and Hovy (1996). The paradigm
of adaptive document generation by “selection and repair”, as introduced by DiMarco,
Hirst, Wilkinson, and Wanner (1995), that is, selection of the relevant pieces of
information from a master document, and then repair of any syntactic or stylistic
problems in the resulting document by a text-repair facility, is central to the goals of a
10 customizable document system. However, the system should be able to support either
an adaptive generation system with full facilities for selecting and repairing texts, as
described by DiMarco, Hirst, Wilkinson, and Wanner (1995) and Hirst, DiMarco,
Hovy, and Parsons (1997), or a simpler version of the system, based on “generation
by selection only”, i.e., with no facilities for textual repair, an implementation of
15 which (called “WebbeDoc”) is described by DiMarco and Foster (1997).

In summary, an author of a customizable document needs to be able to
describe the variations of a document, which may be both textual and non-textual, at
various levels of the document structure, together with the conditions for selecting
each variation.

20 The author then needs a means of selecting all the appropriate variations for a
particular purpose or audience, re-assembling the selected variations into a coherent
document, and producing an appropriately customized version of the document, in
potentially many different levels of representation (e.g., surface English, a deep
syntactic or semantic representation for use in textual repair) and presentation formats
25 (e.g., HTML, LaTeX).

None of the existing adaptive document generation systems has provided a
generally applicable method and apparatus for describing all the different ways in
which a document could be customized, or for providing for a non-computer-
programmer author of a customizable document to specify the possible variations, or
30 for selecting the appropriate variations and producing a customized version of a
document.

Summary of the Invention

It is therefore an object of this invention to provide a system which mitigates to some extent the above outlined disadvantages. Also, the methods used by the present invention are more general than those used in previous systems, allowing not only the potential inclusion of text plans and schemas, text templates, and canned text, but also the dynamic generation of text that can then be subjected to very fine-grained revision and tailoring by a text-repair facility.

This invention seeks to provide a computer system for customizing an initial master document containing information for a multiplicity of versions of the document intended for the different purposes or different users, for a specific purpose or for a specific user.

In accordance with this invention, there is provided a computer system for customizing an initial master document in accordance with a user-defined set of purpose parameters. In accordance with a further aspect of the invention there is provided a data structure (i.e., the customizable, "master", document) for specifying relationships between elements of the document and between elements of the document and their variations.

A master document will therefore contain all the information that the system might need to include in any particular customized version of the document, together with annotations giving the selection conditions as to when each piece of information is relevant and other annotations giving linguistic and formatting information, including for multimedia elements of the document.

A further aspect of the invention provides for a method and apparatus for reading said data structure into a form implementable on a suitably programmed processor such that the implementation of the data structure can store both the form and content of a master document, i.e., all the elements of the document and their variations, and can also act as the process for selecting the relevant variations of the document, according to given values of input parameters specifying the intended purpose or intended user, and then generating the appropriately customized version of the document.

Brief Description of the Drawings

A better understanding of the invention will be obtained by reference to the detailed description of a preferred embodiment below and in regard to the following drawings in which:

- Figure 1 is a schematic diagram showing the architecture of the system;
- 5 Figure 2 shows a generalised form of a data structure for specifying a customizable, master, document according to an embodiment of the present invention;
- Figure 3(a) shows a generalised form of a data substructure for use within the main data structure for specifying linguistic or presentation format information for a component of the master document;
- 10 Figure 3(b) shows a generalised form of a data substructure for use within the main data structure for specifying hypertext links to parts of the main data structure or to other data structures of the form as specified in Figure 2 and provided in other source datafiles;
- Figure 4 is a flowchart of the overall process of reading-in as input the data structure
- 15 and generating as output a final customized version of the document; and
- Figure 5 is a graph showing the resolution process for a customized document generated according to an embodiment of the present invention;

Detailed Description of A Preferred Embodiment

- 20 Referring to Figure 1, an architecture of a customizable document system, the "tailoring engine", is shown generally by numeral 10. The system comprises: a datafile 12 including a data structure 13 for defining a master document; a parser 14 for reading in the contents of the datafile 12 and for creating instances of the document-class data structures 18 in accordance with the general definitions of
- 25 document-class data structures 16; a user input interface 20 for reading new values of purpose parameters 22 which are input by a selection engine 18. The selection engine 18 uses the current values of the purpose parameters to select the relevant variations of each component of the document and to generate appropriately customized versions 26 of a document, which may also include hypertext links to new documents
- 30 28, which may themselves be customizable documents of the form illustrated in Figure 2.

The term "purpose parameter" as used herein means a parameter used in evaluating a selection condition associated with a particular variation of a document

structure, where this parameter can be used in defining either a particular intended purpose or use of a customized version of the document or a particular intended user or audience for a customized version. These elements are explained in detail below.

Furthermore, the datafile may be generated by an authoring tool 30 in accordance with the detailed explanation below.

Referring to figure 2 an embodiment of the data structure 13 according to the present invention is described, which shows the blocks, sub-blocks, fields, and subfields of the data structure.

The data structure 13 has the following main blocks of information:

- 10 1. Identification of purpose parameters and representation-level parameters, and their possible values.
2. Identification of toplevel object (i.e., the main Document).
3. Definitions of main Document and any subDocuments.
4. Definitions of Sections.
- 15 5. Definitions of Topics.
6. Definitions of Sentences.
7. Definitions of Lexicals.
8. Definitions of Words.
9. Definitions of Annotations (replaces previous 9. Definitions of Formats).
- 20 10. Definitions of External objects.

Each of these blocks of the data structure will now be described in turn.

Block 1: The purpose parameters. The first block of the data structure identifies 25 the purpose parameters, or user parameters, together with their possible values, that can be used in forming the Boolean expressions that give the conditions for selecting each variation of a document.

The first block also identifies the representation-level parameters, together with their possible values, that can be used in forming the Boolean expressions that 30 give the conditions for selecting each desired level of representation of the sentences in the master document during the process of generating a customized version of the document.

Block 2: The toplevel object. The toplevel object identifies the document-class instance which is the “root” element of the entire document. It is with this object that the resolution process for generating a customized version of the document begins.

- 5 **Blocks 3-9: The program structures.** Blocks 3-9 describe the program structures, the classes that implement the substructures of the data structure that specify the form and content of a customizable document. The program structures are related in the following manner:

A datafile describing a set of program structures is a particular example of a
10 customizable document created for various uses. The datafile may be divided into various parts.

Firstly, the data structure may be divided into components, or elements, referred to as the classes Document, Section, Topic, Sentence, and Lexical, which each implement a substructure of the data structure that defines a component of a
15 customizable document. Each such substructure of the data structure also includes the variations of a component and the conditions for selecting the appropriate variation of a component.

In addition, the data structure contains basic components which are instances of the classes Word and Annotation, and other components, which are instances of the
20 class External, for linking to other datafiles.

Block 3: The Documents.

In Figure 2, Block 3 describes the instances of the class Document. Each Document description must specify the following properties:

- 25 - A list of its variations. Each variation must be an instance of the class DocumentVariation.
- A list of its annotations. Each annotation must be an instance of the class Annotation.

An instance of the class Annotation can specify both textual and non-textual
30 properties of a document or a component of a document in terms of a particular document-layout format, structure, or linguistic representation. For example, an Annotation object could specify multimedia elements of the document’s layout, such as alignment of text, font size, background colour.

text colour, and graphics; other Annotation objects could specify linguistic information such as discourse relations or coreference links.

Each variation of a Document class is then described as an instance of the class DocumentVariation. Each DocumentVariation description specifies the following properties:

- The condition for selecting this variation. The condition must be a Boolean expression composed from pairs of purpose parameters and their allowable values.
- A list of the components of this variation. Each component must either be an instance of the class Document or an instance of the class Section.
- A list of annotations for this variation. Each annotation must be an instance of the class Annotation.

Block 4: The Sections. Block 4 describes the instances of the class Section. Each Section description specifies the following properties:

- A list of its variations. Each variation must be an instance of the class SectionVariation.
- A list of its annotations. Each annotation must be an instance of the class Annotation.

Each variation of a Section class is then described as an instance of the class SectionVariation. Each SectionVariation description specifies the following properties:

- The condition for selecting this variation. The condition must be a Boolean expression composed from pairs of purpose parameters and their allowable values.
- A list of the components of this variation. Each component must either be an instance of the class Section or an instance of the class Topic.
- A list of the annotations for this variation. Each annotation must be an instance of the class Annotation.

30

Block 5: The Topics. Block 5 describes the instances of the class Topic. Each Topic description specifies the following properties:

- A list of its variations. Each variation must be an instance of the class Topic-Variation.

- A list of its annotations. Each annotation must be an instance of the class Annotation.

Each variation of a Topic class is then described as an instance of the class TopicVariation. Each TopicVariation description specifies the following properties:

- 5 - The condition for selecting this variation. The condition must be a Boolean expression composed from pairs of purpose parameters and their allowable values.
- A list of the components of this variation. Each component must either be an instance of the class Topic or an instance of the class Sentence.
- 10 - A list of the annotations for this variation. Each annotation must be an instance of the class Annotation.

Block 6: The Sentences. Block 6 describes the instances of the class Sentence.

Each Sentence description must specify the following properties:

- 15 - A list of its variations. Each variation must be an instance of the class SentenceVariation.
- A list of its annotations. Each annotation must be an instance of the class Annotation,

Each variation of a Sentence class is then described as an instance of the class Sentence-Variation. Each SentenceVariation description must specify the following properties:

- The condition for selecting this variation. The condition must be a Boolean expression composed from pairs of purpose parameters and their allowable values.
- A list of the components of this variation.
- A list of the representations of this variation. Each representation must be an instance of the class SentenceRepLevel.
- A list of annotations for this variation. Each annotation must be an instance of the class Annotation.

25 30 Each SentenceRepLevel description must specify the following properties:

- The condition for selecting this sentence representation. The condition must be a list of one or more representation-level parameters.
- A list of the components of this variation. Each component must be an instance of the class Lexical.

- A list of the annotations for this variation. Each annotation must be an instance of the class Annotation.

In the current implementation of the system, an instance of the class

5 SentenceRepLevel may be a character string, with any Lexical components identified by surrounding reserved characters. This is a simplification made for ease of testing system prototypes, and does not limit the scope of the invention.

10 **Block 7: The Lexicals.** Block 7 describes the instances of the class Lexical. Each Lexical description must specify the following property:

- A list of its variations. Each variation must be an instance of the class LexicalVariation.
- A list of its annotations. Each annotation must be an instance of the class Annotation.

15 Each variation of a Lexical class is then described as an instance of the class LexicalVariation. Each LexicalVariation description must specify the following properties:

- The condition for selecting this variation. The condition must be a Boolean expression composed from pairs of purpose parameters and their allowable values.
- A character string associated with this LexicalVariation instance.
- A list of annotations for this variation. Each annotation must be an instance of the class Annotation.
- A single component, which may be an instance of any of the classes External, Word, Lexical, Sentence, Topic, Section, or Document. Each of these cases is dealt with as follows:
 1. If a component is a Word, then it is treated as a simple string to be concatenated to the result returned.
 2. If a component is a Lexical, then it is treated as a set of variations of a word, which will be resolved to select the appropriate version of the word. In this way, near-synonymy can be handled within the system.
 3. If a component is a Sentence, Topic, or Section, then it is treated as a set of

variations of a separate piece of a document, which will be resolved to select the appropriate version.

In this way, adaptive hypertext can be handled within the system.

4. If a component is a Document, then it is treated as a whole complete document.

In this way, hypertext links to other documents internal to the same datafile can be handled within the system.

5. If a component is an External object, then it is treated as a whole complete customizable document.

10 In this way, hypertext links to other customizable documents can be handled within the system.

Block 8: The Words. Block 8 describes the instances of the class Word. Each Word description must specify its associated string and its associated annotations.

- 15 Each annotation must be an instance of the class Annotation.

Block 9: The Annotations. Block 9 describes the instances of the class Annotation that will be used to insert all the relevant linguistic and formatting information into the customized version of the document to be output. A description of block 9 is not 20 included in figure 2. Instead, the general structure of an Annotation class object is shown in figure 3(a).

The Annotation objects can be grouped into several distinct sub-taxonomies, one for each type of linguistic or formatting annotations that will be attached to the main master-document data structure. For example, one Annotation sub-taxonomy might specify details of HTML layout for the overall document and each component of the document; another Annotation sub-taxonomy might record properties of the discourse structure of the overall document and each component of the document, such as rhetorical relations and coreference links. Each Annotation object has a property "parent" to reference its immediate ancestor in its (sub-)taxonomy.

30

Block 10: The Externals. Block 10 describes the instances of the class External , which will be used to create hypertext links to other customizable documents specified in other datafiles.

Each External description must specify the following attributes:

- The name of the file containing the external customizable document.
 - A user profile. The user profile is a list of parameters that describe the user or audience for whom the document customization is being performed.
- 5 The general structure of an External class object is shown in figure 3(b).

The data structure 13 according to the present invention allows an author to describe the structure of a customizable document (i.e., a master document). The data structure has a recursive and object-oriented form and can be implemented using an object-oriented programming language so that a customizable document described in 10 the form of the data structure can be implemented as an object-oriented computer program.

The elements of the data structure are related by both part relationships and by inheritance relationships, so that the relationships between the elements of a customizable document described in the form of the data structure and implemented 15 as a corresponding object-oriented computer program can be recognized and maintained by the object hierarchy and inheritance mechanism of an object-oriented computer program.

Moreover, an object-oriented computer program that implements the data structure is both the form and content of a customizable document and the process for 20 selecting and generating an appropriately customized version of the document. Thus the data structure is generic in the sense that it can implement any customizable document given in the form of the data structure.

The data structure 13 describes the corpus text of a specific customizable document (i.e., a master document) having elements and structure, as shown in figures 25 2(a), 2(b) and 2(c) following.

The form of the data structure is defined by the set of general class objects 16 describing the elements and structure of a customizable document in terms of object-oriented program structures. These class objects are related by both part relationships and by inheritance relationships to be explained below.

30 The operation of the system 10 (the "tailoring engine") may be explained as follows:

The parser program 14 reads in and parses the master-document datafile 12 to recognize its structure and then maps the contents of the input datafile into class objects of name according to class names specified in the input data structure.

Properties of the classes are also recognized according to information contained in the data structure.

The parser program 14 also acts as a document-class instantiator program which uses the parsed contents of the input datafile to dynamically create instances of 5 the program structures identified by the general class objects described above.

Properties of the classes are also dynamically assigned according to information contained in the data structure.

An integration of data structures with the main process of the system comprises the following:

- 10 The instances of the class objects dynamically generated from the input datafile, which provide two simultaneous functions:
1. The program data structures describing the elements, structure, and content of the master document.
 2. The selection process for generating the appropriate customized version of 15 the document.

Thus, a feature of the system is that given the current values of the purpose parameters, the instances of the program data structures, that is to say, the instances of the class objects describing the elements of the master document, execute themselves to select and generate the appropriate customized version of the document.

20 The core of the system, that is, the integration of program data structures with the selection process, is generic in terms of the following properties:

25 **Application-independent.** The system core is independent of the application: the only items that need be re-defined for a new application are the input datafile and the interface for reading the current values of the purpose parameters. This is discussed further below.

Platform-independent. The system core is currently implemented in the Java programming language, so is platform-independent to the same extent as Java itself.

30 **Processor-independent.** The system core is currently implemented in the Java programming language, but is independent of the underlying programming language, i.e., processor, to the extent that the programming language used must provide an object-oriented paradigm and a semantics for property inheritance that is consistent with the specification of the resolution process used in the system for generating a customized version of a document from the instances of the program data structures.

The process of generating a customized version of a document from the instances of the program structures is referred to as resolution and will be discussed later with reference to Figure 5.

A customized version of a document can be generated in any number of 5 different levels of representation of its content (e.g., surface English; a syntactic or semantic representation to be used by a text-repair facility; and so on). The different representations to be generated for any given application must be indicated by the representation-level parameters in the description of the document given by the data structure in the input datafile. This information is specified in block 1 of the data 10 structure as described with reference to figures 2(a), 2(b), and 2(c) above.

Each different representation of the content of a customized version of the document will be generated along with a list of all the relevant annotations to the content. These annotations provide information on the multiple forms in which the document may subsequently be presented to a reader, and the linguistic information 15 that may be used to guide subsequent repair of the customized document. The customized document that is generated will also include all annotations to the content concerning the External objects that can be used to provide hypertext links to other customizable documents or to other applications of the system 10 (the "tailoring engine") from figure 1.

20 The process of generating a customized version of a master document is shown in Figure 4. There are two main stages to this process, the initial setup and the main program loop. In the initial setup, the input datafile is read in and parsed, and the appropriate instances of the various document classes described above are created. The parser program 14 reads in the input datafile 12, which contains the data structure 25 13 giving the specification of a customizable document. As it reads in the file, the driver program also acts as a class instantiator to create instances of all relevant document classes according to the input data structure. Program links, i.e., references, are created between these class instances via the setting of their properties and assignment of their property values.

30 In each iteration of the main program loop, new values of the purpose parameters are read in, and a customized version of the document is generated as output for each specified level of representation. A user interface in the form of a reader program 20 obtains the new values of the purpose parameters. In the latter instance, the parameter values may be entered interactively or may be read in from

00000000000000000000000000000000

previously compiled profiles of user preferences stored in computer databases. This allows for the mass customization of information for high volumes of individual users with diverse characteristics, such as in the mass production of personalised financial investment advice.

5 A selection engine 18 resolves the document instances created in the setup stage according to the current values of the purpose parameters to generate the appropriately customized version of the document. The customized document is output in all specified levels of representation with all relevant linguistic and 10 formatting information attached to each component of the document. If there are no more new purpose-parameter values to read in, the main program loop terminates.

The process of selecting the appropriate variation of each document structure is called "resolution". Figure 5 is a graph showing the resolution process for a customized document generated according to an embodiment of the present invention. The pseudocode for the Resolve procedures, which implement the resolution process, 15 is as follows:

Procedure: Resolve (for VariationContainer classes)

Parameters

- (input)

An instance of WorkingCondition

20 (a list of purpose parameters and their current values)

- (input)

An instance of List

(a list of the desired representation levels to output)

- (output)

25 An instance of DocumentObjectSet,

i.e., a set of references to resolved DocumentObjects,

with one set member for each desired level of representation

of this VariationContainer instance

30 Algorithm:

if Resolve has already been called with this WorkingCondition then

result is the previously saved result

else

call Satisfies on each variation of this Variation Container until

```

        the WorkingCondition is satisfied
        call Resolve on this satisfying variation to
            return a DocumentObjectSet
        add the appropriate annotations for this VariationContainer into
5           the returned DocumentObjectSet
        end if
        if no variation satisfies the WorkingCondition then
            result is null
        else
10          result is the annotated DocumentObjectSet for this VariationContainer
        end if

```

Procedure: Resolve (for Variation classes)

Parameters:

```

15      - (input)
            An instance of WorkingCondition
      - (input)
            An instance of List
            (a list of the desired representation levels to output)
20      - (output)
            An instance of DocumentObjectSet,
            i.e., a set of references to resolved DocumentObjects,
            with one set member for each desired level of representation
            of this Variation instance

```

25

Algorithm:

```

create a new DocumentObjectSet for this Variation
for each component of this Variation do
    call Resolve on the current component to
30      return a DocumentObjectSet for this component
        add the appropriate annotations for this Variation into
        the returned DocumentObjectSet
        attach the annotated DocumentObjectSet
        as a child of the DocumentObjectSet for this Variation

```

end for

result is the annotated DocumentObjectSet for this Variation

The "resolution" of the appropriate annotations for a given input data structure is
5 handled as follows:

Each instance of a document class in the input data structure, down to the level
of a Word instance, has an "annotations" property, which may be null, associated with
it. This property is a list of all the Annotation objects that apply to this document-class
instance and must contain one Annotation object for each of the distinct sub-
10 taxonomies in the overall class of Annotation objects.

When a document-class instance is being resolved, its set of Annotation
objects, as specified in its "annotations" property, will be collected and included in
the content of the customized document.

In summary, an application system which wishes to put the invention into
15 practice must execute the following steps:

1. An input datafile of the structure as described with reference to figures
2(a), 2(b), and 2(c) is created;
2. A user interface in the form of a reader program to obtain new values of
purpose parameters is provided. This reader program returns as output an
20 instance of WorkingCondition which is a set of purpose parameters and
their values (implemented as a hashtable in one embodiment of the
invention).
3. The main program first reads in the datafile, then calls the reader program
to obtain a new instance of WorkingCondition. The main program will
25 then start the resolution process for the toplevel Document object by
passing it the WorkingCondition.
4. The output of each iteration of the main program will be a customized
version of the document in all the levels of representation specified in the
input datafile using the representation-level parameters. Each
30 representation of the customized document is output for possible later
processing by the application system.

References

- Giuseppe Carenini, Vibhu O. Mittal, and Johanna D. Moore. "Generating patient-specific interactive natural language explanations." Proceedings, Eighteenth Annual Symposium on Computer Applications in Medical Care, Washington D.C., November 1994, 5-9.
- Alison Cawsey, Kim Binsted, and Ray Jones. "Personalised explanations for patient education." Proceedings of the Fifth European Workshop on Natural Language Generation, 1995, 59-74.
- Chrysanne DiMarco and Mary Ellen Foster. "The automated generation of Web documents that are tailored to the individual reader." Proceeding of the 1997 AAAI Spring Symposium on Natural Language Processing for the World Wide Web, Stanford University, March 1997.
- Chrysanne DiMarco, Graeme Hirst, and Eduard Hovy. "Generation by selection and repair as a method for adapting text for the individual reader." Proceedings of the Workshop on Flexible Hypertext, Eighth ACM International Hypertext Conference, Southampton UK, April 1997.
- Chrysanne DiMarco, Graeme Hirst, Leo Wanner, and John Wilkinson. "HealthDoc: Customizing patient information and health education by medical condition and personal characteristics." Workshop on Artificial Intelligence in Patient Education, Glasgow, August 1995.
- Graeme Hirst, Chrysanne DiMarco, Eduard Hovy, and Kimberley Parsons. "Authoring and generating health-education documents that are tailored to the needs of the individual patient." In: Anthony Jameson, Cécile Paris, and Carlo Tasso (editors), *User Modeling: Proceedings of the Sixth International Conference, UM97* (Chia Laguna, Sardinia, Italy), Vienna and New York: Springer Wien New York, June 1997, 107-118.
- Eduard Hovy and Leo Wanner. "Managing sentence planning requirements." Proceedings, ECAI-96 Workshop on Gaps and Bridges: New Directions in Planning and Natural Language Generation, Budapest, August 1996.
- Alistair Knott, Chris Mellish, Jon Oberlander, and Mick O'Donnell. "Sources of flexibility in dynamic hypertext generation" Proceedings, Eighth International Natural Language Generation Workshop, Herstmonceaux Castle, June 1996, 151-160.
- Maria Milosavljevic and Robert Dale. "Strategies for comparison in encyclopaedia descriptions." Proceedings, Eighth International Natural Language Generation Workshop, Herstmonceaux Castle, UK, June 1996, 161-170.
- Ehud Reiter, Chris Mellish, and John Levine. "Automatic generation of technical documentation." Applied Artificial Intelligence, 9, 1995, 259-287.
- Victor J. Strecher, Matthew Kreuter, Dirk-Jan Den Boer, Sarah Kobrin, Harm J. Hospers, and Celette S. Skinner. "The effects of computer-tailored smoking cessation

messages in family practice settings." *The Journal of Family Practice*, 39(3), September 1994, 262-270.

- ⁵ Leo Wanner and Eduard Hovy. "The HealthDoc sentence planner." Proceedings of the Eighth International Workshop on Natural Language Generation, Brighton, UK, June 1996.

Sample Data Structures

5 Selections from two sample data structures are presented in this section. The first example is for the customizable home page of the HealthDoc Project at the University of Waterloo (Waterloo, Canada). The second example is for a master document giving basic health information on diabetes.

10 Note that in the LexicalVariation labelled “lexDiabetesMaster-a” in Example 1, there is a link to another completely separate customizable document, contained in a different input datafile. This link is made through an instance of the External class.

Example 1: A Customizable Web Page

```

15 // The parameters
PurposeParameters
|role=CExpert physician layperson funder&
technical =high low&
20 age=senior adult child&
formality=formal informal&
coolness=cool bland|
RepresentationLevelParameters
25 |levels=english|
// The top-level object
toplevel=Document.webbedoc
30 // The Documents and DocumentVariations
Document webbedoc
|title="The HealthDoc Project Home Page"&
variations=doc-a doc-b doc-c doc-d doc-e doc-f doc-g doc-h&
annotations=html-doc-webbedoc-toplevel|
35 DocumentVariation doc-a
|condition=(and (coolness cool) (age adult) (role CExpert))&
componentList=Section.sec1 Section.sec2 Section.sec3 Section.sec4
Section.sec5 Section.sec6 Section.sec7&
40 annotations=html-doc-webbedoc-doc-a|
DocumentVariation doc-d
|condition=(and (coolness bland) (age adult))&
componentList=Section.sec1 Section.sec2 Section.sec3 Section.sec4
Section.sec5 Section.sec6 Section.sec7&
45 annotations=html-doc-webbedoc-doc-d|
// The Sections and SectionVariations
Section sec1
50 |variations=sec1a sec1b sec1c sec1d sec1e sec1f

```

```

sec1g sec1h sec1i sec1j sec1k sec1l
sec1m sec1n sec1o sec1p sec1q sec1r&
annotations=html-sec-webbedoc-sec1|
```

5 SectionVariation sec1a
|condition=
 (and (role funder)(technical all)(coolness bland)(formality formal))&
 componentList=Section.subsec1-1 Section.subsec1-2&
 annotations=html-sec-webbedoc-sec1a|

10 Section subsec1-1
|variations=subsec1-1a&
 annotations=html-sec-webbedoc-subsec1-1|

15 SectionVariation subsec1-1a
|condition=()&
 componentList=Topic.topic1&
 annotations=html-sec-webbedoc-subsec1-1|

20 Section sec2
|variations=sec2a sec2b&
 annotations=html-sec-webbedoc-sec2|

25 Section subsec2-1
|variations=subsec2-1a&
 annotations=html-sec-webbedoc-subsec2-1|

30 SectionVariation subsec2-1a
|condition=()&
 componentList=Topic.topic4 Topic.topic5 Topic.topic6 Topic.topic7&
 annotations=html-sec-webbedoc-subsec2-1|

// The Topics and TopicVariations

Topic topic4
35 |variations=topic4a&
 annotations=html-topic-webbedoc-default|

TopicVariation topic4a
|condition=()&
40 componentList=Sentence.sent4a-1 Sentence.sent4a-2&
 annotations=html-topic-webbedoc-default|

Topic topic5
45 |variations=topic5a topic5b topic5c topic5d&
 annotations=html-topic-webbedoc-default|

TopicVariation topic5a
|condition=(and (role physician) (technical high))&
50 componentList=Sentence.sent5a-1&

```
annotations=html-topic-webbedoc-default|  
  
TopicVariation topic5c  
|condition=(and (not (role physician)) (technical high))&  
5 componentList=Sentence.sent5c-1&  
annotations=html-topic-webbedoc-default|  
  
Topic topic7  
|variations=topic7a topic7b&  
10 annotations=html-topic-webbedoc-default|  
  
TopicVariation topic7a  
|condition=(technical low)&  
componentList=Sentence.sent7a-1&  
15 annotations=html-topic-webbedoc-default|  
  
TopicVariation topic7b  
|condition=(technical high)&  
componentList=Sentence.sent7b-1&  
20 annotations=html-topic-webbedoc-default|  
  
Topic topic-compliance  
|variations=topic-compliance-a&  
annotations=html-topic-webbedoc-default|  
25  
TopicVariation topic-compliance-a  
|condition=()&  
componentList=Sentence.sent-compliance-1 Sentence.sent-compliance-2  
Sentence.sent-compliance-3 Sentence.sent-compliance-4&  
30 annotations=html-topic-webbedoc-default|  
  
// The Sentences and SentenceVariations  
Sentence sent4a-1  
35 |variations=sent4a-1i&  
annotations=html-sent-webbedoc-default|  
  
SentenceVariation sent4a-1i  
|condition=()&  
40 componentList=()&  
levelList=SentenceRepLevel.sent4a-1i-english&  
annotations=html-sent-webbedoc-default|  
  
SentenceRepLevel sent4a-1i-english  
45 |repLevel=english&  
componentList="Why do we want to be able to produce tailored  
documents?"&  
annotations=html-sent-webbedoc-default|
```

Sentence sent4a-2
|variations=sent4a-2i&
 annotations=html-sent-webbedoc-default|

5 SentenceVariation sent4a-2i
|condition=()&
 componentList=()&
 levelList=SentenceRepLevel.sent4a-2i-english&
 annotations=html-sent-webbedoc-default|

10 SentenceRepLevel sent4a-2i-english
|repLevel=english&
 componentList="Because research in communication has shown that
 people pay more attention to messages that are aimed
 just at them."&
 annotations=html-sent-webbedoc-default|

15 Sentence sent5a-1
|variations=sent5a-1i&
 annotations=html-sent-webbedoc-default|

SentenceVariation sent5a-1i
|condition=()&
 componentList=()&
25 levelList=SentenceRepLevel.sent5a-1i-english&
 annotations=html-sent-webbedoc-default|

SentenceRepLevel sent5a-1i-english
|repLevel=english&
30 componentList="Studies have shown that health information that is tailored
 to a patient's specific medical condition and personal characteristics
 is much more effective than generic information in influencing
 ^lexCompliance^ and subsequent outcome."&
 annotations=html-sent-webbedoc-default|

35 Sentence sent7a-1
|variations= sent7a-1i&
 annotations=html-sent-webbedoc-default|

40 SentenceVariation sent7a-1i
|condition=()&
 componentList=()&
 levelList=SentenceRepLevel.sent7a-1i-english&
 annotations=html-sent-webbedoc-default|

45 SentenceRepLevel sent7a-1i-english
|repLevel=english&

componentList="But it can be very difficult to write and keep track
of many versions of the same ^lexSynonyms1^."&
annotations=html-sent-webbedoc-default|

5 Sentence sent8a-1
|variations=sent8a-1i&
annotations=html-sent-webbedoc-default|

SentenceVariation sent8a-1i
10 |condition=()&
componentList=()
levelList=SentenceRepLevel.sent8a-1i-english&
annotations=html-sent-webbedoc-default|

15 SentenceRepLevel sent8a-1i-english
|repLevel=english&
componentList="What is needed is a computer system for the production
of tailored health-information and patient-education
documents, that would, on demand, customize a 'master
20 document' to the needs of a particular individual."&
annotations=html-sent-webbedoc-default|

Sentence sent8a-2
25 |variations=sent8a-2i&
annotations=html-sent-webbedoc-default|

SentenceVariation sent8a-2i
30 |condition=()&
componentList=()
levelList=SentenceRepLevel.sent8a-2i-english&
annotations=html-sent-webbedoc-default|

SentenceRepLevel sent8a-2i-english
35 |repLevel=english&
componentList="The HealthDoc project has currently built the first
^lexDiabetesMaster^ of such a system."&
annotations=html-sent-webbedoc-default|

40 Sentence sent-compliance-1
|variations=sent-compliance-1a&
annotations=html-sent-webbedoc-default|

45 SentenceVariation sent-compliance-1a
|condition=()&
componentList=()
levelList=SentenceRepLevel.sent-compliance-1a-english&
50 annotations=html-sent-webbedoc-default|

5 SentenceRcpLevel sent-compliance-1a-english
|repLevel=english&
componentList="Recent experiments have shown that health-education
material can be much more effective if it is
customized for the individual reader in accordance
with their medical conditions, demographic variables,
personality profile, or other relevant factors."&
annotations=html-sent-webbedoc-default|

10 Sentence sent-compliance-2
|variations=sent-compliance-2a&
annotations=html-sent-webbedoc-default|

15 SentenceVariation sent-compliance-2a
|condition=()&
componentList=()&
levelList=SentenceRepLevel.sent-compliance-2a-english&
annotations=html-sent-webbedoc-default|

20 SentenceRepLevel sent-compliance-2a-english
|repLevel=english&
componentList="For example, Dr Victor Strecher (now at the
Comprehensive Cancer Center of the University of
Michigan) and colleagues sent unsolicited leaflets to
patients of family practices on topics such as giving
up smoking, improving dietary behaviour, or having a
mammogram."&
annotations=html-sent-webbedoc-default|

25 Sentence sent-compliance-3
|variations=sent-compliance-3a&
annotations=html-sent-webbedoc-default|

30 SentenceVariation sent-compliance-3a
|condition=()&
componentList=()&
levelList=SentenceRepLevel.sent-compliance-3a-english&
annotations=html-sent-webbedoc-default|

35 SentenceRepLevel sent-compliance-3a-english
|repLevel=english&
componentList="Each leaflet was 'tailored' to the recipient, on the
basis of data gathered from them in an earlier survey."&
annotations=html-sent-webbedoc-default|

40 Sentence sent-compliance-4
|variations=sent-compliance-4a&
annotations=html-sent-webbedoc-default|

45 Sentence sent-compliance-4
|variations=sent-compliance-4a&
annotations=html-sent-webbedoc-default|

SentenceVariation sent-compliance-4a
|condition=()&
componentList=()&
levelList=SentenceRepLevel.sent-compliance-4a-english&
5 annotations=html-sent-webbedoc-default|

SentenceRepLevel sent-compliance-4a-english
|repLevel=english&
componentList="In each study, the 'tailored' leaflets were found to
10 have a significantly greater effect on the patients'
behaviour than 'generic' leaflets had upon patients
in a control group."&
annotations=html-sent-webbedoc-default|

15

// The Lexicals and LexicalVariations

Lexical lexSynonyms1
|variations=lexSynonyms1a lexSynonyms1b&
20 annotations=|

LexicalVariation lexSynonyms1a
|condition=(role physician)&
string="brochure"&
25 value=Word.brochure&
annotations=|

LexicalVariation lexSynonyms1b
|condition=(not (role physician))&
30 string="information"&
value=Word.information&
annotations=|

Lexical lexCompliance
35 |variations=lexCompliance-a&
annotations=|

LexicalVariation lexCompliance-a
|condition=(role physician)&
40 string="tailored health-information"&
value=Topic.topic-compliance&
annotations=|

Lexical lexDiabetesMaster
45 |variations=lexDiabetesMaster-a&
annotations=|

```
LexicalVariation lexDiabetesMaster-a
|condition=()&
string="prototype"&
value=External.external-diabetes&
5 annotations=|
// The Words
Word brochure
|value="brochure"|
10 Word information
|value="information"|
// The Annotations
15 Annotation html-doc-webbedoc-toplevel
title="The HealthDoc Home Page"
Annotation html-doc-webbedoc-doc-a
|background="blue2.gif"&
20 bgcolor="#ffffef"&
vlink="#990099"&
alink="#990099"&
link="#990099"&
title-begin=<center><font size = +4 color = \'990099\'>"&
25 title-end=</font></center>"&
image-dir="/~healthdo/images/"&
image-align="alt"&
parent=html-doc-webbedoc-toplevel|
30 Annotation html-sec-webbedoc-sec1
|title-begin=<font size = +1 color = \'990099\'>"&
title-end=</font>"&
title="The goal of the HealthDoc project"&
parent=html-doc-webbedoc-toplevel|
35 Annotation html-sec-webbedoc-sec1a
|image="businmen.jpg"&
parent=html-sec-webbedoc-sec1|
40 Annotation html-sec-webbedoc-sec2
|title-begin=<font size = +1 color = \'990099\'>"&
title-end=</font>"&
title="The motivation for the research"&
parent=html-doc-webbedoc-toplevel|
45 // The External objects
External external-diabetes
|fileName="diabetes.master"&
profile=<list of parameters describing current user/audience>|
50
```

Example 2: Customizable Health Information

```

// The parameters
5 PurposeParameters
| type=insulin-dependent non-insulin-dependent&
technical=high-technical moderate-technical low-technical&
age=senior adult young-adult child&
locus-of-control=doctor patient|


10 RepresentationLevelParameters
|english spl|


// The top-level object
15 toplevel=Document.diabetes


// The Documents and DocumentVariations
Document diabetes
20 |title="Treating Your Diabetes"&
variations=doc-a&
annotations=html-doc-diabetes-toplevel discourse-doc-diabetes-toplevel|


25 DocumentVariation doc-a
|condition=()&
componentList=Section.sec1 Section.sec2 Section.sec3 Section.sec4&
annotations=html-doc-diabetes-doc-a discourse-doc-diabetes-toplevel|


// The Sections and SectionVariations
30 Section sec1
|variations=sec1a&
annotations=html-sec-diabetes-sec1 discourse-sec-diabetes-sec1|


35 SectionVariation sec1a
|condition=()&
componentList=Section.subsec1-1 Section.subsec1-2&
annotations=html-sec-diabetes-sec1a discourse-sec-diabetes-sec1|


40 Section subsec1-1
|variations=subsec1-1a&
annotations=html-sec-diabetes-subsec1-1 discourse-sec-diabetes-sec1|


45 SectionVariation subsec1-1a
|condition=()&
componentList=Topic.topic1&
annotations=html-sec-diabetes-subsec1-1a discourse-sec-diabetes-sec1|


50 Section subsec1-2
|variations=subsec1-2a&
annotations=html-sec-diabetes-subsec1-2 discourse-sec-diabetes-sec1|

```

SectionVariation subsec1-2a
|condition=()&
componentList=Topic.topic2 Topic.topic3 Topic.topic4
Topic.topic5 Topic.topic6&
5 annotations=html-sec-diabetes-subsec1-2a discourse-sec-diabetes-sec1|

// The Topics and TopicVariations
Topic topic1
|variations=topic1a&
10 annotations=html-topic-diabetes-default discourse-topic-diabetes-topic1|

TopicVariation topic1a
|condition=()&
componentList=Sentence.sent1a-1 Sentence.sent1a-2&
15 annotations=html-topic-diabetes-default discourse-topic-diabetes-topic1|

Topic topic3
|variations=topic3a topic3b&
20 annotations=html-topic-diabetes-default discourse-topic-diabetes-topic3|

TopicVariation topic3a
|condition=(type insulin-dependent)&
componentList=Sentence.sent3a-1&
25 annotations=html-topic-diabetes-default discourse-topic-diabetes-topic3|

TopicVariation topic3b
|condition=(type non-insulin-dependent)&
componentList=Sentence.sent3b-1&
30 annotations=html-topic-diabetes-default discourse-topic-diabetes-topic3|

// The Sentences and SentenceVariations
Sentence sent1a-1
|variations=sent1a-1i&
35 annotations=html-sent-diabetes-default discourse-sent-diabetes-default|

SentenceVariation sent1a-1i
|condition=()&
componentList=()&
40 levelList=SentenceRepLevel.sent1a-1i-english SentenceRepLevel.sent1a-1i-spl&
annotations=html-sent-webbedoc-default|

SentenceRepLevel sent1a-1i-english
|repLevel=english&
45 componentList=Lexical.lexDiab Lexical.lexis Lexical.lexa Lexical.lexgroup
Lexical.lexof Lexical.lexconds Lexical.lexin Lexical.lexwhich
Lexical.lexglucose Lexical.lexlevels Lexical.lexare
Lexical.lexabnormally Lexical.lexhigh.&
50 annotations=html-sent-diabetes-default discourse-sent-diabetes-default|

```

SentenceRepLevel sent1a-1i-spl
|repLevel=spl&
componentList="<a Sentence Plan Language (SPL) form>"&
annotations=html-sent-diabetes-default discourse-sent-diabetes-default|
5
Sentence sent3a-1
|variations=sent3a-1i&
annotations=html-sent-diabetes-default discourse-sent-diabetes-default|
10 SentenceVariation sent3a-1i
|condition=()&
componentList=()&
levelList=SentenceRepLevel.sent3a-1i-english SentenceRepLevel.sent3a-1i-spl&
annotations=html-sent-webbedoc-default|
15 SentenceRepLevel sent3a-1i-english
|repLevel=english&
componentList="The condition that you have is insulin-dependent diabetes."&
annotations=html-sent-diabetes-default discourse-sent-diabetes-default|
20 SentenceRepLevel sent3a-1i-spl
|repLevel=spl&
componentList="(asc / ascription
:tense present
25 :domain (cond1 / abstraction
:lex condition
:determiner the
:process (have / ownership
:lex have-possession
30 :tense present
:domain (hearer / person)
:range cond))
:range (diab2 / abstraction
:lex diabetes
35 :determiner zero
:property-ascription (ins / quality
:lex insulin-dependent)))"&
annotations=html-sent-diabetes-default discourse-sent-diabetes-default|
// The Lexicals and LexicalVariations
40 Lexical lexDiab
|variations=lexDiab-a&
annotations=|
LexicalVariation lexDiab-a
45 |condition=()&
string="Diabetes"&
value=Word.Diab&
annotations=|

```

```

Lexical lexglucose
|variations=lexglucose-a&
 annotations=|
5   LexicalVariation lexglucose-a
|condition=()
 string="glucose"&
 value=Word.glucose&
 annotations=|
10  Lexical lexhigh
|variations=lexhigh-a&
 annotations=|
15  LexicalVariation lexhigh-a
|condition=()
 string="high"&
 value=Word.high&
 annotations=|
20 // The Words
Word Diab
|value=="Diabetes"&|
25 Word glucose
|value=="glucose"|

Word high
|value=="high"&|
30

// The Annotations

35 // // HTML Annotations
Annotation html-doc-diabetes-toplevel
|title="About Your Diabetes"|

Annotation html-doc-diabetes-doc-a
40 |bgcolor="#ffffef"&
 title-begin="<h1 align=\"center\">"&title-end="</h1>"&
 parent=html-doc-diabetes-toplevel|


Annotation html-sec-diabetes-default
45 // <default HTML markup for any Section in diabetes document>|


Annotation html-sec-diabetes-sec1
|title="Basic information"&
 parent=html-sec-diabetes-default|
50

```

Annotation html-sec-diabetes-sec1a
|title-begin="<h2 align='center'>"& title-end="</h2>"&
parent=html-sec-diabetes-sec1|

5 Annotation html-sec-diabetes-subsec1-1
|title="What is diabetes?"&
parent=html-sec-diabetes-sec1|

10 Annotation html-sec-diabetes-subsec1-1a
|section-end="<p>"&
title-begin="<h3>"&title-end="</h3>"&
parent=html-sec-diabetes-subsec1-1|

15 Annotation html-sec-diabetes-subsec1-2
|title="The two types of diabetes"&
parent=html-sec-diabetes-sec1|

20 Annotation html-sec-diabetes-subsec1-2a
|section-end="<p>"&
title-begin="<h3>"&title-end="</h3>"&
parent=html-sec-diabetes-subsec1-2|

// // Linguistic Annotations

Annotation discourse-doc-diabetes-toplevel
|title="About Your Diabetes"|

Annotation discourse-sec-diabetes-sec1
|title="Basic information"&
relations="((ord < (topic2a topic3a))
30 (ord < (topic2a topic3b))
(ord < (topic2a topic4a))
(ord < (topic2a topic5a))
(ord < (topic2a topic5b))
(ord < (topic2a topic5c))
35 (ord < (topic2a topic5d))
(ord < (topic2a topic6a))
(ord < (topic2a topic6b))
(ord < (topic3a topic6a))
(ord < (topic3b topic6b))
40 (elaboration topic4a topic3b)
(justification topic2a topic3a)
(justification topic2a topic3b)
(justification topic2a topic4a)
(elaboration topic4a topic2a)
45 (elaboration topic5a topic3a)
(elaboration topic5b topic3b)
(elaboration topic5c topic3a)
(elaboration topic5d topic3b)
(justification topic3a topic6a)
50 (justification topic3b topic6b)"|

Annotation discourse-sec-diabetes-subsec1-1
|title="What is diabetes?"&
parent=discourse-sec-diabetes-sec1|

5 Annotation discourse-sec-diabetes-subsec1-2
|title="The two types of diabetes"&
parent=discourse-sec-diabetes-sec1|

10 Annotation discourse-topic-diabetes-topic3
|corefs="((cond1 specific cond)
 (diab2 specific diab)
 (cond1 generic diab2))"&
parent=discourse-sec-diabetes-sec1|

15

The Pseudocode

The program objects are:

- BasicObject.
- 5 ResolvableObject.
- DocumentObject.
- DocumentObjectSet.
- VariationContainer.
- Variation.
- 10 Condition.
- WorkingCondition.
- Annotation.
- External.

15

The Procedures

Procedures used by BasicObject classes

- These procedures are used within BasicObject classes (e.g., BasicDocument,
20 BasicSection, etc.)

Procedure: SetProperties

Parameter:

A list of property names and their corresponding values

25

Algorithm:

```
for each property in the list do  
    set its value
```

30 Procedures used by VariationContainer classes

These procedures are used within VariationContainer classes (e.g., Document, Section, Topic), which have a list of variations. VariationContainer classes are extensions of ResolvableObject and therefore have a Resolve method.

Procedure: SetProperties

Parameter:

A list of property names and their corresponding values

Algorithm:

5 for each property in the list do
 if property is a list of variations then
 set the class property "variations" to given value
 else if property is a list of annotations then
 set the class property "annotations" to given value
 else if property is defined for this class then
 set its value
 else signal an error
 end if
 end for

15

Procedure: Resolve (for VariationContainer classes)

Parameters

- (input)

An instance of WorkingCondition

20 (a list of purpose parameters and their current values)

- (input)

An instance of List

(a list of the desired representation levels to output)

- (output)

25 An instance of DocumentObjectSet,

i.e., a set of references to resolved DocumentObjects,

with one set member for each desired level of representation

of this VariationContainer instance

Algorithm:

30 if Resolve has already been called with this WorkingCondition then
 result is the previously saved result
 else
 call Satisfies on each variation of this Variation Container until
 the WorkingCondition is satisfied

call Resolve on this satisfying variation to
 return a DocumentObjectSet
add the appropriate annotations for this VariationContainer into
 the returned DocumentObjectSet
5 end if
 if no variation satisfies the WorkingCondition then
 result is null
 else
 result is the annotated DocumentObjectSet for this VariationContainer
10 end if

Procedures used by Variation classes

These procedures are used within Variation classes (e.g., DocumentVariation, SectionVariation), which therefore have a selection condition.

15 **Procedure: SetProperties**

Parameters: A list of property names and their corresponding values

Algorithm:

for each property in the list do
 if property is a condition then
 set the class property “condition” to given value
 else if property is a list of annotations then
 set the class property “annotations” to given value
 else if property is a list of components then
 set the class property “componentList” to given value
20
25
 else if property is defined for this class then
 set its value
 else signal an error
 end if
30 end for

Procedure: Satisfies

Parameters:

- (input)
 - An instance of WorkingCondition
 - (output)
 - A Boolean value

Algorithm:

```
if this variation satisfies the WorkingCondition then  
    result is true  
else  
    result is false  
end if
```

Procedure: Resolve (for Variation classes)

Parameters:

- (input)
 - An instance of WorkingCondition
 - (input)
 - An instance of List
 - (a list of the desired representation levels to output)
 - (output)
 - An instance of DocumentObjectSet,
 - i.e., a set of references to resolved DocumentObjects,
 - with one set member for each desired level of representation
 - of this Variation instance

Algorithm:

create a new DocumentObjectSet for this Variation

for each component of this Variation do

call `Resolve` on the current component to

return a DocumentObjectSet for this component

add the appropriate annotations for this Variation into

the returned DocumentObjectSet

attach the annotated DocumentObjectSet

as a child of the DocumentObjectSet for this Variation

end for
result is the annotated DocumentObjectSet for this Variation

Toplevel Procedures

5 These are the toplevel procedures used to read in a datafile containing
a master document, create instances of the document-class objects, then loop to read
in new values of the purpose parameters and generate an appropriately customized
version of the document, with all appropriate annotations, at each level of
representation as specified by the representation-level parameters given in the
10 datafile.

Procedure: Parse

Parameter:

Name of datafile to be read in

15 Algorithm:
while end-of-file has not been reached
read in the next line of the datafile

20 if the line is a comment or a blank line then
skip over it
else if the line specifies the toplevel DocumentObject then
set the "toplevel" variable to reference this object
else if the line specifies the purpose parameters and their values then
set the "purposeParameters" variable
else if the line specifies the possible representation levels then
set the "repLevels" variable
else instantiate the specified document object
(i.e., create a new executable instance of this document-object class,
30 set a reference to this instance,
assign a name to this instance,
assign its properties)
end of while loop

return the reference to the toplevel DocumentObject,
the list of purpose parameters and their possible values,
the list of desired representation levels

5 **Procedure:** Reader

Algorithm:

call an application-specific interface to:
- read in current values of purpose parameters
- read in desired representation levels
- 10 create an instance of WorkingCondition using these values

Procedure: Main

Parameter:

Name of the input datafile

15 Algorithm:

call Parse
call Reader
while new purpose parameters are input
 call Resolve on the toplevel object
 for each specified level of representation
 output the corresponding DocumentObjectSet
 end for
 call Reader
end of while loop

25

The present application has been described with reference to a presently preferred embodiment. Modifications and variations of that embodiment will be apparent to a person of skill in the art. Such modifications and variants are believed to be within the scope of the present invention as defined in the claims appended hereto.

30

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A computer system for generating custom versions of a master document in accordance with user purpose parameters, said system comprising:
 - 5 (a) a datafile including a data structure specifying said master document;
 - (b) a document class library for providing general document class definitions;
 - (c) a parser for creating executable instances of said document-class data structures from said datafile;
 - (d) a user input interface for inputting said purpose parameters; and
 - 10 (e) a selection engine for generating said custom versions of said document from said instances of said document class library by utilizing current values of said purpose parameters.
- 15 2. A computer system as defined in claim 1, said document including a plurality of elements and said data structure defining relationships between said elements and variations thereof.
- 20 3. A computer system as defined in claim 1, said instances including a plurality of elements and variations each arranged in a hierarchy of predetermined relationships.
- 25 4. A computer system as defined in claim 3, each element including a resolver for selecting from its variations and each variation including a resolver for expanding said variation into its component document objects.
5. A computer system as defined in claim 3, said elements being the components of said document structure.
- 30 6. A computer system as defined in claim 1, said instances each including linguistic information for providing automated grammatical and stylistic correction of said customized documents.
7. A computer readable memory device encoded with a data structure for generating custom versions of a master document, the data structure having a plurality of

general class objects for describing elements and variations providing, structure and content of said master document.

8. A device as defined in claim 6, said data structure including:
5 (a) a first data block for identifying user purpose parameters for selecting versions of said master document;

(b) a toplevel object for identifying a document-class instance; and

(c) a program structure for specifying the form and content of said custom
versions of said master document.

10

9. A device as defined in claim 7, said data structure including an external object for
describing instances of an external class for defining links to other master
documents to thereby to create a network of systems for generating custom
documents.

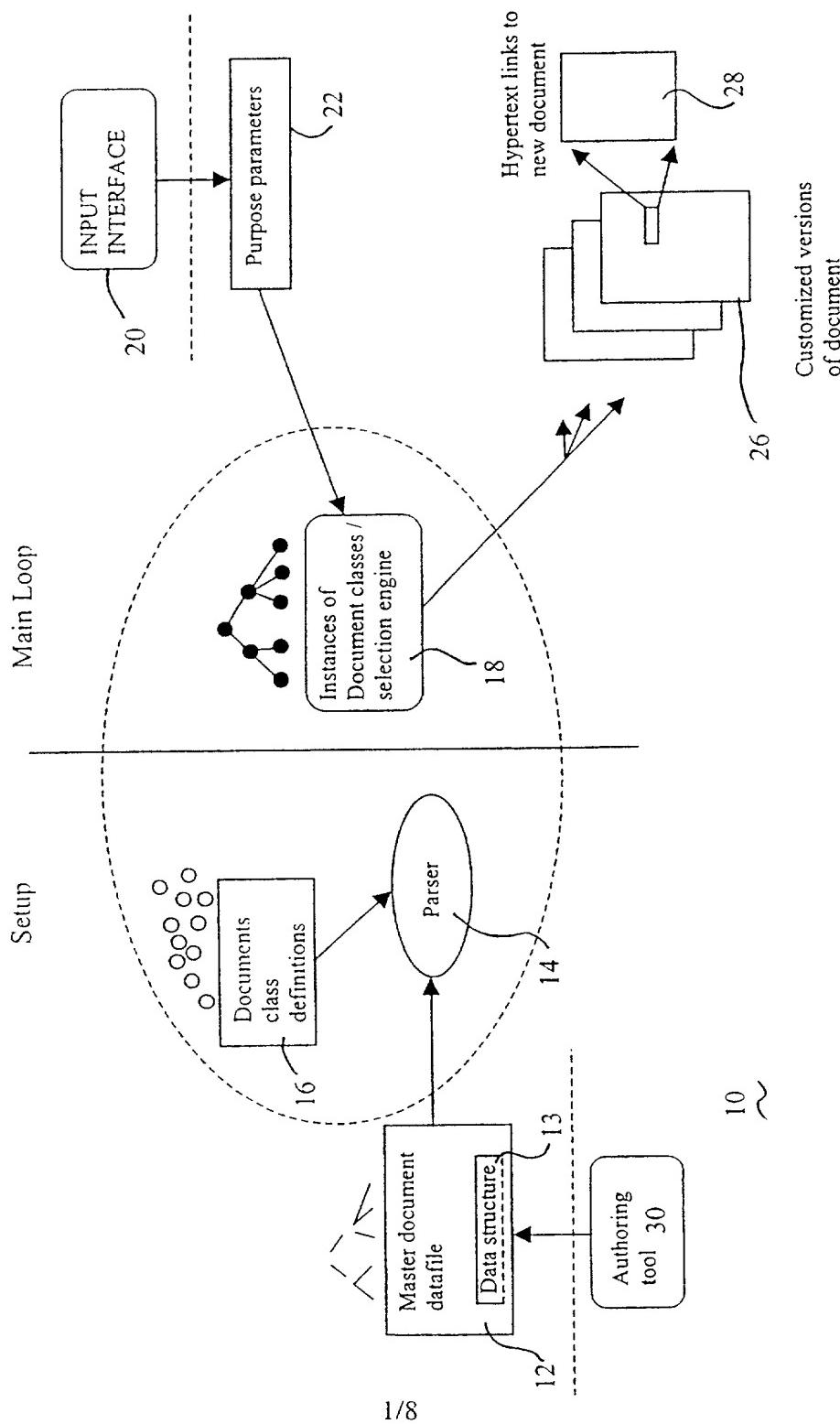
15

10. A device as defined in claim 8, said links being hypertext links.

11. A method for generating custom versions of a master document implemented in a
computer system and in accordance with user purpose parameters, said method
20 comprising the steps of:

- (a) specifying said master document having a datafile and including a data
structure;
- (b) providing a document class library having general document class definitions;
- (c) parsing said datafile for creating executable instances of said document-class
25 data structures;
- (d) a user input interface for inputting said purpose parameters; and using current
values of said purpose parameters to generate custom versions of said
document from said instances of said document class library.

- 30 12. A method as defined in claim 11, including executing said instance for selecting
from variations of an element and expanding each variation into its component
document objects, wherein each said instances includes a plurality of elements and
variations each arranged in a hierarchy of predetermined relationships.

**FIGURE 1**

Block 1—The parameters:

```

PurposeParameters
  (<feature 1>=<list of possible values>)&
  (<feature 2>=<list of possible values>)&
  [Subsequent lines define remaining features]

```

```

RepresentationLevelParameters
  (<list of representation levels to output>)

```

Block 2—Identification of the top-level object:

```
toplevel=Document.<name of main Document>
```

Block 3—Definitions of all Documents and their variations:

Sub-block: Definition of a Document and its variations.

Field: Definition of the Document.

```

Document <name of document>
|variations=<list of variations of the document>&
|annotations=<list of annotations>|

```

Field: Definitions of the Document's variations.

Subfield: Definition of a variation of the Document.

```

DocumentVariation <name of document variation>
|condition=<Boolean expression>&
|componentList=<list of components of the document variation>&
|annotations=<list of annotations>|

```

Subsequent subfields define remaining variations of the Document.

Subsequent sub-blocks define remaining Documents and their variations.

Block 4—Definitions of all Sections and their variations:

Sub-block: Definition of a Section and its variations

Field: Definition of the Section.

```

Section <name of section>
|variations=<list of variations of the section>&
|annotations=<list of annotations>|

```

Field: Definitions of the Section's variations.

Subfield: Definition of a variation of the Section.

```

SectionVariation <name of section variation>
|condition=<Boolean expression>&
|componentList=<list of components of the section variation>&
|annotations=<list of annotations>|

```

Subsequent subfields define remaining variations of the Section.

Subsequent sub-blocks define remaining Sections and their variations.

FIGURE 2 (a)

Block 5—Definitions of all Topics and their variations:

Sub-block: Definition of a Topic and its variations.

Field: Definition of the Topic.

```
Topic <name of topic>
|variations=<list of variations of the topic>&
|annotations=<list of annotations>|
```

Field: Definitions of the Topic's variations.

Subfield: Definition of a variation of the Topic.

```
TopicVariation <name of topic variation>
|condition=<Boolean expression>&
|componentList=<list of components of the topic variation>&
|annotations=<list of annotations>|
```

Subsequent subfields define remaining variations of the Topic

Subsequent sub-blocks define remaining Topics and their variations.

Block 6—Definitions of all Sentences and their variations:

Sub-block: Definition of a Sentence and its variations.

Field: Definition of the Sentence.

```
Sentence <name of sentence>
|variations=<list of variations of sentence>&
|annotations=<list of annotations>|
```

Field: Definitions of the Sentence's variations.

Subfield: Definition of a variation of the Sentence.

```
SentenceVariation <name of sentence variation>
|condition=<Boolean expression>&
|componentList=<list of components>&
|levelList=<list of representation levels>&
|annotations=<list of annotations>|
```

Subfield: Definitions of the Sentence's representations.

```
SentenceRepLevel <name>
|repLevel=<list of representation levels>&
|componentList=<a string or list of components>&
|annotations=<list of annotations>|
```

Subsequent subfields define remaining variations of the Sentence and their representations

Subsequent sub-blocks define remaining Sentences and their variations

FIGURE 2(6)

Block 7—Definitions of all Lexicals and their variations:

Sub-block: Definition of a Lexical and its variations.

Field: Definition of the Lexical.

```
Lexical <name of lexical item>
|variations=<list of variations of lexical item>&
|annotations=<list of annotations>|
```

Field: Definitions of the Lexical's variations

Subfield: Definition of a variation of the Lexical.

```
LexicalVariation <name of lexical variation>
|condition=<Boolean expression>&
|string=<character string>&
|value=<DocumentClass>.<objectName>&
|annotations=<list of annotations>|
```

Subsequent subfields define remaining variations of the Lexical.

Subsequent sub-blocks define remaining Lexicals and their variations.

Block 8—Definitions of all Words:

Field: Definition of a Word.

```
Word <word>
|value="<>word>"|
```

Subsequent fields define remaining Words.

FIGURE 2(c)

```
Annotation <name of the Annotation object>
  {<property>=<value>"&
   <property>=<value>"&
   <Subsequent lines define remaining properties and their values>&
   parent=<name of the immediate ancestor of this Annotation object>|
```

FIGURE 3(a)

```
External <name of the External object>
|fileName=<file name>&
|profile=<list of parameters describing current user/audience>|
```

FIGURE 3C6)

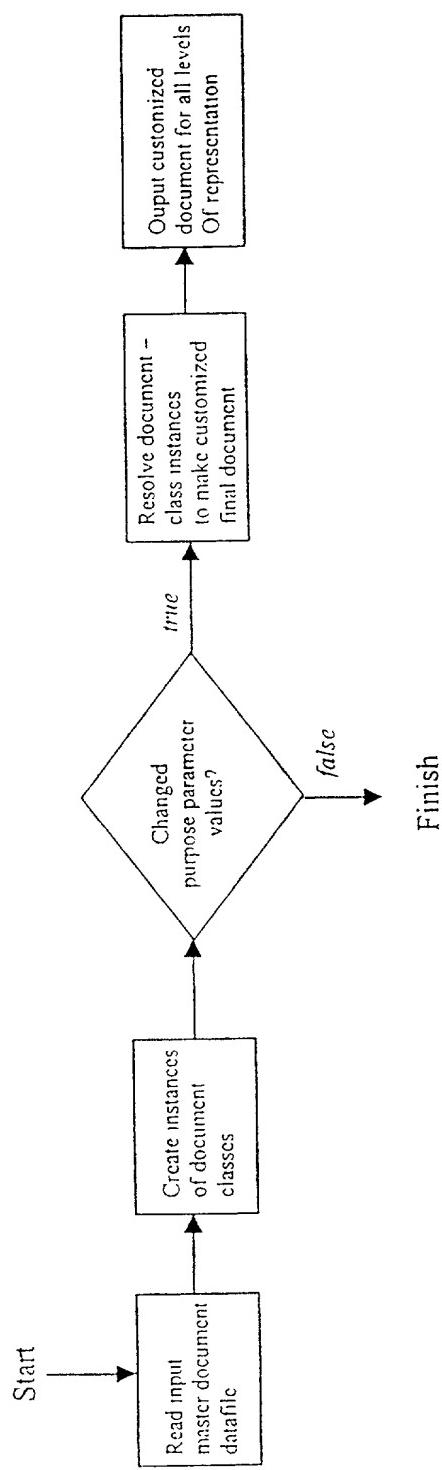
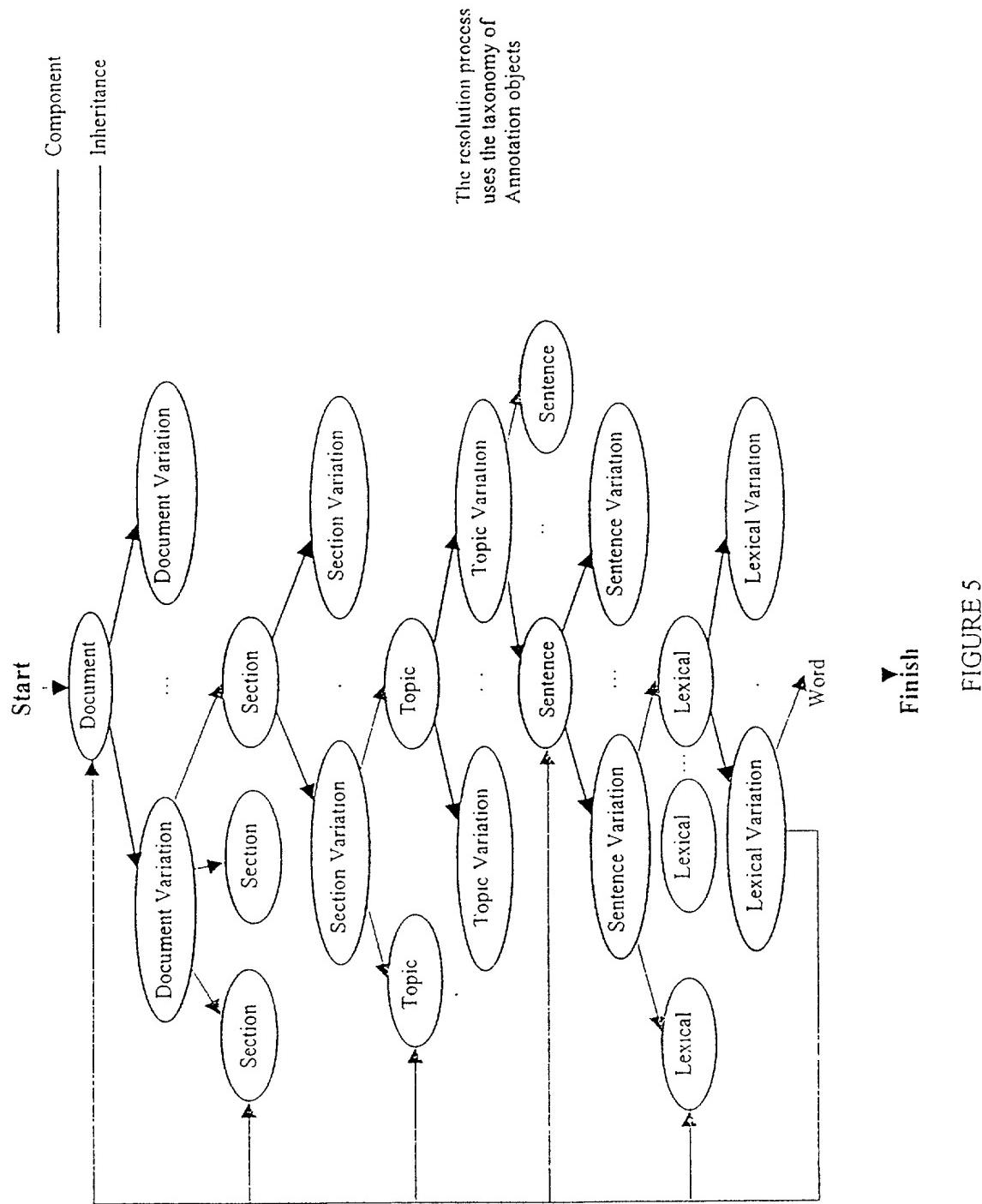


FIGURE 4



• Docket No.
8700137-0006

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD AND APPARATUS FOR AUTHORIZING OF CUSTOMIZABLE MULTIMEDIA DOCUMENTS

the specification of which

(check one)

is attached hereto.

was filed on _____ as United States Application No. or PCT International Application Number _____
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Priority Not Claimed

9716986.6

United Kingdom

11 August 1997

(Number)
9720133.9

(Country)
United Kingdom

(Day/Month/Year Filed)
22 September 1997

(Number)
2,230,367

(Country)
Canada

(Day/Month/Year Filed)
24 February 1998

(Number)

(Country)

(Day/Month/Year Filed)

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

I hereby claim the benefit under 35 U. S. C. Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, C. F. R., Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

PCT/CA98/00771

11 August 1998

pending

(Application Serial No.)

(Filing Date)

(Status)

(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)

(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)

(patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (*list name and registration number*)

Kevin Pillay	41,559
John R.S. Orange	29,725
Santosh K. Chari	41,477

Send Correspondence to: **Orange Chari Pillay**
Suite 3600, P.O. Box 190
Toronto Dominion Bank Tower, Toronto Dominion Centre
Toronto, Ontario M5K 1H6 Canada

Direct Telephone Calls to: (*name and telephone number*)
Kevin Pillay (416) 868-3457

Full name of sole or first inventor DIMARCO, Chrysanne	Date
Sole or first inventor's signature	
Residence 78 Bideford Avenue, Toronto, Ontario M3H 1K4 Canada	
Citizenship Canadian	
Post Office Address 78 Bideford Avenue, Toronto, Ontario M3H 1K4 Canada	

Full name of second inventor, if any FOSTER, Mary Ellen	Date
Second inventor's signature	
Residence Apartment 2508, 666 Spadina Avenue, Toronto, Ontario M5S 2H8 Canada	
Citizenship Canadian	
Post Office Address Apartment 2508, 666 Spadina Avenue, Toronto, Ontario M5S 2H8 Canada	

IN THE EUROPEAN PATENT OFFICE

Applicant: **DIMARCO, Chrysanne et al.**

Serial No.: **PCT/CA98/00771**

Title: **A Method And Apparatus For Authoring
Of Customizable Multimedia Documents**

Filed: **August 11, 1998**

Our Ref.: **8700137-0005**

October 7, 1999

International Preliminary Examination Authority
European Patent Office
Erhardstrasse 27
D-80331 Munich
Germany

Attention: G. Barraco, Examiner

RESPONSE

Dear Sir:

This is in response to the First Written Opinion dated May 21, 1999.

The Examiner has indicated that claims 1 to 12 lack inventive step in view of reference D1: C. Dimarco et al.: "The automated generation of Web documents that are tailored for the reader".

The Examiner has indicated that the customization of documents for different audiences and purposes is known in the art as indicated and implementing such a method in an object oriented approach is obvious.

The Applicant has carefully reviewed the Examiner's statements and the D1 reference. Applicant respectfully believes that the Examiner's grounds for rejecting the claims are unfounded, particularly in light of the following comments.

The present invention defines a computer system and method for generating custom versions of a master document. The master document is defined by a data file including a data structure which describes the content of the master document, a document class library which contains generic descriptions for the various versions of the document, a parser which reads both the data file and the document class library to create executable instances of the document and a selection engine which uses purpose parameters to generate custom versions

of the document from the various instances of the executable document as generally defined in independent claims 1 and 11.

On the contrary the D1 reference describes a system for customizing documents wherein the master document is a script file that has to be run each time the purpose parameters are changed. There is no hint of suggestion in D1 of having a document class library containing generic descriptions. In D1, the data file which is the master document is reread whenever new purpose parameters are provided by the user. There is no concept of instantiation in D1 as required and defined in the claims of the present invention.

Furthermore in D1 there is no suggestion of a selection engine, a parser or an object orientated class library. The process of instantiation is important in the present invention in that it provides a “program” that holds its state which is then subsequently processed by a selection engine for producing the customized document in response to the user purpose parameters. On the other hand, in D1, the master document is a script file that has to be run each time the purpose parameters are changed. Thus there is no disclosure of a generic master document which is content independent. The D1 reference defines merely instances of a master document which is content dependent.

The Examiner has indicated that it is “obvious, when implementing the program design of such methods, to adopt the object or approach according to now a days trends”. The Applicant has carefully reviewed the D1 reference and can find no suggestion or indication of adopting an object orientated approach to create customized documents based on the data file of D1. The Examiner has also failed to provide any other indication of the state of the art regarding object oriented approaches with regard to customized document generation. Thus it is believed that the Examiner’s objection is not well founded and is merely the result of hindsight analysis.

For the above reasons it is believed that the objections to independent claims 1 and 11 are overcome and consequently the objections to the dependent claims 2 to 10 and 12 are also overcome.

Reconsideration of the claims on their merits is courteously requested.

Respectfully submitted,
ORANGE CHARI PILLAY

Agents for the Applicant
per:


Kevin Pillay
Tel: (416) 868-3457

KP/st

DRAFT - DO NOT CITE OR RELY UPON